Linguaggi e Traduttori – Tempo: 2 ore

Prof. Marco Gavanelli

20 dicembre 2017

Esercizio 1 (3 punti)

Si consideri il linguaggio $L = \{a^n b^n | n > 0\} \cup \{a^n b^k | n, k \leq 1\}$.

Si classifichi il linguaggio secondo Chomsky.

Si scriva una grammatica non ambigua che genera il linguaggio L; si fornisca la grammatica di livello più basso possibile nella classificazione secondo Chomsky (intendendo il livello 3 come minimo e il livello 0 come massimo).

Se è possibile, si mostri l'albero di derivazione della stringa $aabb$.

Esercizio 2 (6 punti)

Si consideri la grammatica $G = \langle \{a, b, c, d, f\}, \{S, A, B\}, P, S \rangle$, dove:

$$P = \begin{array}{l} S \to AdB \mid ABd \\ A \to bB \mid Af \\ B \to aB \mid Aa \mid c \end{array}$$

1. Si classifichi la grammatica secondo Chomsky.
2. La grammatica è LL(1)? Se sì, si scriva la parsing table del PDA riconoscitore. Se no, si motivi il perché.
3. La grammatica è LR(0)? Se sì, si scriva l'automa a stati finiti ausiliario del PDA riconoscitore. Se no, si motivi il perché.

Qualora nei punti precedenti si sia riusciti a ottenere uno o due riconoscitori deterministici, si mostri il riconoscimento delle stringhe $bcfdbca$ e $bcbcfc$ mostrando l'evoluzione dello stack.

Esercizio 3 (3 punti)

Sia data la seguente grammatica: $G = \langle \{a, b\}, \{A, B, C\}, P, A \rangle$, dove

$$P = \begin{array}{l} A \to bB \mid aC \\ B \to a \mid aA \mid bC \\ C \to b \mid aC \end{array}$$

Si scriva un'espressione regolare equivalente, motivando i passaggi.

Soluzione 1

Il linguaggio è di tipo 2.

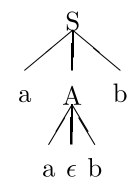
Si noti come la stringa ab sia in comune ad entrambi gli insiemi con cui il linguaggio è definito: per avere una grammatica non ambigua bisognerà evitare che la stringa ab venga generata da più produzioni.

Una possibile grammatica:

$G = \langle \{a, b\}, \{S, A, B\}, P, S \rangle$

$$P = \begin{array}{l} S \to A \mid a \mid b \\ A \to aAb \mid \epsilon \end{array}$$

L'albero di derivazione:

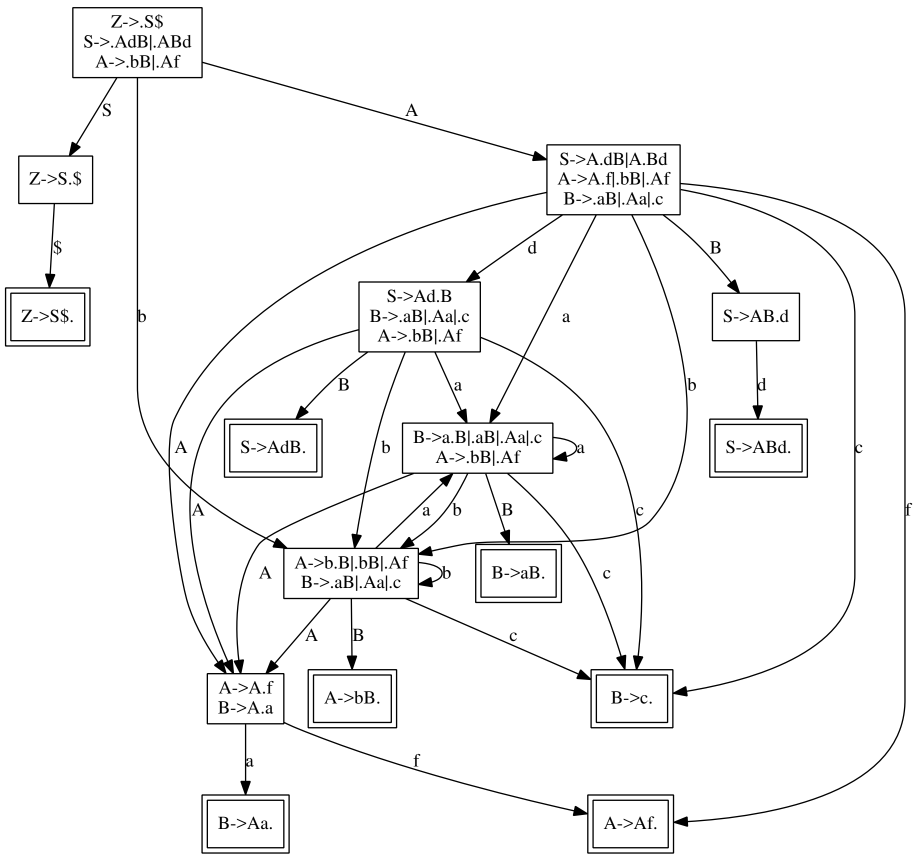


Soluzione 2

1. La grammatica è di tipo 2 (context-free).

2. La grammatica non è LL(1), come si vede dal fatto che c'è una ricorsione sinistra per A.

3. La grammatica è LR(0), come si può vedere dal fatto che l'automa non presenta conflitti:


Z->.S$
S->.AdB|.ABd
A->.bB|.Af
Z->S.$
Z->S$.
S->A.dB|A.Bd
A->A.f|.bB|.Af
B->.aB|.Aa|.c
S->Ad.B
B->.aB|.Aa|.c
A->.bB|.Af
S->AB.d
S->AdB.
B->a.B|.aB|.Aa|.c
A->.bB|.Af
S->ABd.
A->b.B|.bB|.Af
B->.aB|.Aa|.c
B->aB.
A->A.f
B->A.a
A->bB.
B->c.
B->Aa.
A->Af.

Riconoscimento:

Input	Stack
bcfdbca$	
cfdbca$	b
fdbca$	bc
fdbca$	bB
fdbca$	A
dbca$	Af
dbca$	A
bca$	Ad
ca$	Adb
a$	Adbc
a$	AdbB
a$	AdA
$	AdAa
$	AdB
$	S
	S$
	Z

Stringa riconosciuta.

Input	Stack
bcbcfc$	
cbcfc$	b
bcfc$	bc
bcfc$	bB
bcfc$	A
cfc$	Ab
fc$	Abc
fc$	AbB
fc$	AA
c$	AAf
c$	AA

Stringa non riconosciuta.

Soluzione 3

Riscriviamo le produzioni della grammatica in termini di equazioni

$$\begin{cases} A = bB + aC \\ B = a + aA + bC \\ C = b + aC \end{cases}$$

Eliminiamo la ricorsione diretta per C e sostituiamo per C:

$$\begin{cases} A = bB + aa^*b \\ B = a + aA + ba^*b \\ C = a^*b \end{cases}$$

Sostituiamo A nell'equazione per B:

$$\begin{cases} A = bB + aa^*b \\ B = a + abB + aaa^*b + ba^*b \end{cases}$$

Eliminiamo la ricorsione diretta per B:

$$\begin{cases} A = bB + aa^*b \\ B = (ab)^*(a + aaa^*b + ba^*b) \end{cases}$$

Sostituiamo in A: $L(A) = b(ab)^*(a + aaa^*b + ba^*b) + aa^*b$